


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

## Structure and chance: melding logic and probability for software debugging

Full text Pdf (4.65 MB)

**Source** [Communications of the ACM archive](#)  
 Volume 38 , Issue 3 (March 1995) [table of contents](#)  
 Pages: 31 - ff.  
 Year of Publication: 1995  
 ISSN:0001-0782

**Authors** [Lisa](#) Department of Computer Science and Engineering, The University of Texas at Arlington, 416 Yates Street,  
[Burnell](#) Nedderman Hall, Room 300, Arlington, TX  
[Eric](#)  
[Horvitz](#) Microsoft Research, One Microsoft Way, Redmond, WA

**Publisher** ACM Press New York, NY, USA

**Additional Information:** [abstract](#) [references](#) [citing](#) [index terms](#) [review](#) [collaborative colleagues](#) [peer to peer](#)

**Tools and Actions:** [Discussions](#) [Find similar Articles](#) [Review this Article](#)  
[Save this Article to a Binder](#) [Display Formats:](#) [BibTex](#) [EndNote](#) [ACM Ref](#)

**DOI Bookmark:** Use this link to bookmark this Article: <http://doi.acm.org/10.1145/203330.203338>  
[What is a DOI?](#)

**Warning:** The download time has expired please click on the item to try again.

### ↑ ABSTRACT

Software errors abound in the world of computing. Sophisticated computer programs rank high on the list of the most complex systems ever created by humankind. The complexity of a program or a set of interacting programs makes it extremely difficult to perform offline verification of run-time behavior. Thus, the creation and maintenance of program code is often linked to a process of incremental refinement and ongoing detection and correction of errors. To be sure, the detection and repair of program errors is an inescapable part of the process of software development. However, run-time software errors may be discovered in fielded applications days, months, or even years after the software was last modified—especially in applications composed of a plethora of separate programs created and updated by different people at different times. In such complex applications, software errors are revealed through the run-time interaction of hundreds of distinct processes competing for limited memory and CPU resources. Software developers and support engineers responsible for correcting software problems face difficult challenges in tracking down the source of run-time errors in complex applications. The information made available to engineers about the nature of a failure often leaves open a wide range of possibilities that must be sifted through carefully in searching for an underlying error.

### ↑ REFERENCES